UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Explorando as Características Texturais em Imagens Biomédicas: Avaliação de GLCM e VGG16 para Classificação de Grupos Tratados e controles

Laura Nicoleti Zamproni

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP
SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Explorando Características Texturais em Imagens Biomédicas: Avaliação de GLCM e VGG16 para Classificação de Grupos Tratados e Controles

Laura Nicoleti Zamproni

USP - São Carlos

2024

Explorando Características Texturais em Imagens Biomédicas:
Avaliação de GLCM e VGG16 para Classificação de Grupos Tratados e Controles

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.
Área de concentração: Inteligência Artificial.
Orientador: Odemir Bruno

USP - São Carlos

2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Ze Zamproni, Laura Nicoleti
Explorando Características Texturais em Imagens Biomédicas: Avaliação de GLCM e VGG16 para Classificação de Grupos Tratados e Controles / Laura Nicoleti Zamproni; orientador Odemir Martinez Bruno. -- São Carlos, 2024.
51 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. classificação imagens. 2. imagens de microscopia de fluorescência . 3. inteligênga artificial. 4. GLCM. 5. VGG16. I. Bruno, Odemir Martinez, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Resumo

Zamproni, L.N. **Explorando Características Texturais em Imagens Biomédicas: Avaliação de GLCM e VGG16 para Classificação de Grupos Tratados e Controles**. 2024. 47p. Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade Federal de São Paulo, São Carlos, 2024.

A análise de imagens biomédicas é uma ferramenta essencial para investigar processos biológicos e identificar padrões associados a condições normais ou patológicas. No entanto, o volume crescente de dados gerados por tecnologias modernas de imagem desafia os métodos tradicionais em termos de precisão, escalabilidade e velocidade. A integração de inteligência artificial e aprendizado de máquina pode superar essas limitações, proporcionando análises mais robustas e automatizadas. Neste estudo, aplicamos abordagens computacionais para investigar alterações morfológicas em células-tronco neurais (NSCs) tratadas com hemoglobina. Foram utilizados dois métodos de extração de características, GLCM e VGG16, cuja eficácia foi avaliada por análise de componentes principais (PCA) e classificadores baseados em SVM e MLP. Os resultados mostraram que a separação entre os grupos "Controle" e "Tratado" era evidente nos gráficos de PCA, mesmo quando diferenças não eram detectáveis visualmente. Enquanto o GLCM destacou mudanças texturais locais, o VGG16 capturou padrões globais abstratos, ambos consistentes com alterações morfológicas associadas ao estresse oxidativo e inflamatório induzido pelo tratamento. Os achados corroboram dados de expressão gênica que sugerem perda de pluripotência e transição para um fenótipo glial, refletindo remodelações no citoesqueleto e na matriz extracelular. Este estudo destaca o potencial das técnicas computacionais avançadas na biologia celular, contribuindo para novas perspectivas no estudo de processos patológicos e no desenvolvimento de terapias regenerativas.

Palavras-chave: Células-tronco neurais, Inteligência artificial, Aprendizado de máquina, GLCM, VGG16, PCA, SVM, MLP.

Abstract

Zamproni, L.N. **Exploring Textural Features in Biomedical Images: Evaluation of GLCM and VGG16 for Classification of Treated and Control Groups**.2024. 47p. Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade Federal de São Paulo, São Carlos, 2024.

Biomedical image analysis is an essential tool for investigating biological processes and identifying patterns associated with normal or pathological conditions. However, the increasing volume of data generated by modern imaging technologies challenges traditional methods in terms of precision, scalability, and speed. The integration of artificial intelligence and machine learning can overcome these limitations, enabling more robust and automated analyses. In this study, we applied computational approaches to investigate morphological changes in neural stem cells (NSCs) treated with hemoglobin. Two feature extraction methods, GLCM and VGG16, were used, and their effectiveness was evaluated using principal component analysis (PCA) and classifiers based on SVM and MLP. The results showed that the separation between the "Control" and "Treated" groups was evident in the PCA plots, even when differences were not visually detectable. While GLCM highlighted local textural changes, VGG16 captured more abstract global patterns, both consistent with morphological changes associated with oxidative and inflammatory stress induced by the treatment. The findings align with gene expression data suggesting the loss of pluripotency and a transition to a glial phenotype, reflecting remodeling of the cytoskeleton and extracellular matrix. This study highlights the potential of advanced computational techniques in cell biology, contributing to new perspectives in the study of pathological processes and the development of regenerative therapies.

Keywords: Neural stem cells, Artificial intelligence, Machine learning, GLCM, VGG16, PCA, SVM, MLP.

Lista de Figuras

Figura 1. Os classificadores de imagens baseados em DL podem interpretar com precisão as alterações na morfologia celular na triagem de alto rendimento baseada em imagens. 16
Figura 2. Exemplos de imagens dos grupos Controle e Tratado.....22
Figura 3. PCA das características extraídas da GLCM. 23
Figura 4.Matriz de Confusão do classificador SVM das características extraídas pela GCLM.....24
Figura 5.Matriz de Confusão do classificador MLP das características extraídas pela GCLM......25
Figura 6.PCA das características extraídas da VGG16.....26
Figura 7. Matriz de Confusão do classificador SVM das características extraídas pela VGG16.....27
Figura 8. Matriz de Confusão do classificador MLP das características extraídas pela VGG16......28
Figura 9. Arquitetura da VGG16......31

Sumário

Resumo 6

Abstract 7

1 Introdução 10

2 Revisão Bibliográfica 12

2.1 Aprendizado de máquina e aprendizado profundo 12

2.2 Aplicações biológicas de DL 13

2.2.1 Classificação de imagens. 13

2.3 O problema biológico deste estudo 14

3 Proposta do Trabalho 17

4 Materiais e métodos 18

4.1 Aquisição de imagens 18

4. 2 Classificação das imagens 18

4.3 Pipelines 20

5 Resultados 21

5.1 Imagens 21

5.2 Características Extraídas e Análise de Padrões Texturais - GLCM 23

5.3 Classificadores aplicados a GLCM 23

5.4 Características Extraídas – VGG16 25

5.5 Classificadores aplicados à VGG16 26

6. Discussão 29

7 Conclusão 34

REFERÊNCIAS 35

Apêndce I Códigos 39

1 Introdução

A análise de imagens biomédicas desempenha um papel crucial tanto na pesquisa quanto na prática clínica, viabilizando a investigação detalhada de processos biológicos e a identificação de padrões associados a condições normais ou patológicas. Desde a detecção de estruturas celulares até a segmentação de órgãos em exames de imagem, a interpretação precisa de dados visuais é indispensável para o avanço da biomedicina (BEARER, 2003). Contudo, com o aumento exponencial na geração de dados provenientes de tecnologias modernas de imagem, os métodos tradicionais enfrentam desafios significativos, como a necessidade de maior precisão, velocidade e escalabilidade para lidar com grandes volumes de informações (PENG; ZHOU; ZHOU; BRIA *et al.*, 2016).

Historicamente, a análise manual de imagens de microscopia, conduzida por especialistas, era a principal abordagem utilizada. Esse método envolve a observação direta de padrões, identificação de estruturas específicas e medições manuais de características, como dimensões e distâncias. Embora útil em pequenos conjuntos de dados ou para validações pontuais, a análise manual apresenta limitações consideráveis, incluindo o viés subjetivo do operador, o tempo excessivo demandado e a incapacidade de processar grandes volumes de imagens com eficiência. Essas características tornam a abordagem inadequada para estudos que exigem alta precisão e escalabilidade (MEZEI; KOLCSáR; JOó; GURZU, 2024).

Nesse cenário, as tecnologias de inteligência artificial (IA), com destaque para o aprendizado de máquina (ML) e o aprendizado profundo (DL), emergem como ferramentas revolucionárias para a análise de imagens biomédicas. A IA automatiza o processo de análise ao empregar redes neurais profundas que aprendem padrões diretamente dos dados brutos, possibilitando a construção de modelos altamente precisos e eficientes (LI; ZHANG; YANG; TENG, 2024). Redes neurais convolucionais (CNNs), por exemplo, têm sido amplamente utilizadas para tarefas como classificação, segmentação e rastreamento de objetos em imagens biomédicas. Essas técnicas se destacam pela capacidade de detectar detalhes sutis frequentemente imperceptíveis ao olho humano, ampliando a acurácia e a confiabilidade das análises (GU; WANG; KUEN; MA *et al.*, 2018).

Neste trabalho, utilizaremos ferramentas de big data e inteligência artificial para classificar imagens de células-tronco neurais expostas ou não à hemoglobina, mimetizando os efeitos de uma hemorragia cerebral. Essa abordagem integrará métodos avançados de análise

com modelos biológicos complexos, contribuindo para uma compreensão mais aprofundada das respostas celulares nesse contexto patológico.

2 Revisão Bibliográfica

2.1 Aprendizado de máquina e aprendizado profundo

O aprendizado de máquina (do inglês, machine learning, ML) é agora a chave para a IA e a maneira fundamental de tornar os computadores inteligentes, permitindo que as máquinas reconheçam padrões e tomem decisões com o mínimo de intervenção humana (PUSHPANATHAN; HANAFI; MASHOHOR; FAZLIL ILAHI, 2021). Na verdade, ML é uma combinação de ciência da computação e estatística (CLARKE; PARMESAR; SALEEM; RAMANAN, 2022). O ML permite que os programas processem grandes quantidades de dados sem depender da modificação manual dos parâmetros(JORDAN; MITCHELL, 2015).

Aplicações mais práticas de ML são abundantes em diagnóstico clínico, terapia de precisão, triagem de drogas, monitoramento de saúde e muitas outras configurações (GOECKS; JALILI; HEISER; GRAY, 2020). O ML geralmente é composto por um extrator de recursos e um classificador. Primeiro, é necessário um especialista com conhecimento em vários domínios para projetar um extrator de recursos que possa analisar quais dados são os recursos mais importantes e convertê-los em uma combinação de vetores de recursos. Em seguida, a máquina aprende esses vetores de recursos e encontra os padrões correspondentes. Por fim, o classificador pode detectar ou classificar automaticamente os padrões de entrada com base nos vetores de recursos de entradas (DU; CHEN; LI; YANG *et al.*, 2023).

O DL é um subcampo do ML que usa uma cascata de unidade de processamento não linear multicamada (ou seja, redes neurais) para extração e transformação de recursos e realiza representação de recursos multinível e aprendizado de conceito abstrato (DU; CHEN; LI; YANG *et al.*, 2023). O DL não exige que especialistas projetem manualmente a engenharia de recursos complexos, mas transfere diretamente os dados para as entradas da rede neural, que podem facilmente obter treinamento de ponta a ponta e mostram melhores vantagens com big data. Como resultado, é mais flexível do que os modelos tradicionais de aprendizado de máquina, como árvores de decisão (DT), regressão logística (LR) e máquinas de vetores de suporte (SVMs)(WU; JI; ZHAO; JI *et al.*, 2016).

Com o rápido desenvolvimento da IA e DL uma série de algoritmos excelentes foi produzida. Atualmente, esses algoritmos baseados em DL estão sendo aplicados para estudar imagens biológicas, ajudando os biólogos a obterem a análise e interpretação dos dados de imagem (MOEN; BANNON; KUDO; GRAF *et al.*, 2019). A DL inclui muitos tipos diferentes de redes neurais. Na análise de dados biomédicos, as redes mais comumente usadas são a rede

neural profunda (do inglês deep neural network, DNN), rede neural convolucional (do inglês convolutional neural network, CNN), rede neural recorrente (do inglês recurrent neural network, RNN) e rede adversária generativa (do inglês generative adversarial network, GAN)(DU; CHEN; LI; YANG *et al.*, 2023). A CNN é atualmente a rede mais amplamente utilizada, mais completa e tem bom desempenho na classificação, segmentação e detecção de objetos de imagens. De um modo geral, a CNN consiste em uma camada de entrada, várias camadas convolucionais, várias camadas de agrupamento, uma ou mais camadas totalmente conectadas e uma camada de saída (KRIZHEVSKY; SUTSKEVER; HINTON, 2017).

2.2 Aplicações biológicas de DL

Existem quatro uso críticos do DL em biomedicina: classificação de imagens, segmentação de imagens, rastreamento de objetos e microscopia aumentada.(MOEN; BANNON; KUDO; GRAF *et al.*, 2019). A seguir discutiremos a classificação de imagens (Figura 1).

2.2.1 Classificação de imagens.

A classificação de imagens é a tarefa de atribuir um rótulo significativo a uma imagem e foi um dos primeiros sucessos de alto perfil do DL. Um exemplo clássico é a discriminação entre imagens de cães e gatos; um exemplo biológico seria identificar se uma proteína é expressa no citoplasma ou no núcleo com base na fluorescência (MOEN; BANNON; KUDO; GRAF *et al.*, 2019).

Devido à utilidade da classificação de imagem, muito do trabalho recente em visão computacional se concentrou em melhorar o desempenho em conjuntos de dados padrão, como ImageNet(HE; ZHANG; REN; SUN; HUANG; LIU; VAN DER MAATEN; WEINBERGER, 2017). Como as arquiteturas das imagens biológicas são muito semelhantes, se não iguais, as imagens convencionais, e à relativa falta de dados de treinamento para imagens biológicas, o aprendizado de transferência tem destaque na criação de classificadores de imagens com bom desempenho em dados biológicos(PAWLOWSKI; CAICEDO; SINGH; CARPENTER *et al.*, 2016; ZHANG; LI; ZENG; SUN *et al.*). Pode-se implementar o aprendizado de transferência nesses casos começando com um classificador de imagem que foi treinado em um grande conjunto de dados, como ImageNet, substituindo a camada final por uma adequada para a nova tarefa de classificação e, em seguida, treinando novamente em um conjunto menor de dados anotados(ZHANG; LI; ZENG; SUN *et al.*).

As aplicações de classificação de imagens baseada em DL para dados biológicos demonstram as vantagens técnicas do DL para descoberta biológica. Por exemplo, a maioria dos trabalhos sobre a interpretação de imagens concentrou-se na geração de classificadores que identificam condições e compostos que levam a mudanças significativas na morfologia celular (GODINEZ; HOSSAIN; LAZIC; DAVIES *et al.*, 2017). Para explicar as mudanças na morfologia celular que podem não ser capturadas em dados rotulados, várias abordagens usaram modelos de aprendizado profundo para extrair vetores de recursos em vez de rótulos e, em seguida, agrupar esses vetores (Figura 1)(KANDASWAMY; SILVA; ALEXANDRE; SANTOS, 2016; PAWLOWSKI; CAICEDO; SINGH; CARPENTER *et al.*, 2016). Os classificadores de imagem também foram usados para identificar alterações no estado celular: em um estudo recente, os cientistas usaram um marcador fluorescente de diferenciação para estabelecer uma verdade básica e depois treinaram um classificador para identificar células diferenciadas diretamente de imagens de campo claro (BUGGENTHIN; BUETTNER; HOPPE; ENDELE *et al.*, 2017). Esses estudos destacam o fato de que o aprendizado profundo é uma ferramenta acessível que pode ajudar os biólogos a entenderem seus dados de imagem.

2.3 O problema biológico deste estudo

A neurogênese, o processo de geração de novos neurônios a partir de células-tronco neurais (NSCs), desempenha um papel essencial no desenvolvimento do sistema nervoso e na manutenção da plasticidade cerebral ao longo da vida (ALVAREZ-BUYLLA; GARCIA-VERDUGO, 2002; RIBEIRO; XAPELLI, 2021). Em condições normais, as NSCs apresentam a capacidade de se autorrenovar e diferenciar em neurônios, astrócitos e oligodendrócitos, garantindo a homeostase do tecido neural (ALVAREZ-BUYLLA; GARCIA-VERDUGO, 2002). No entanto, em contextos patológicos, como após hemorragias cerebrais, esse equilíbrio pode ser significativamente perturbado, levando a consequências prejudiciais para a função cerebral (STęPIEń; TARKA; CHUTORAńSKI; FELCZAK *et al.*, 2018).

A hemorragia cerebral, incluindo a subaracnoide e intraventricular, resulta na liberação de produtos derivados do sangue, como a hemoglobina, no microambiente neural. A presença de hemoglobina e seus produtos de degradação induz estresse oxidativo, inflamação e alterações no microambiente extracelular, criando condições adversas para a neurogênese (DRVENICA; STANčIć; MASLOVARIć; TRIVANOVIć *et al.*, 2022; RIFKIND; MOHANTY; NAGABABU, 2014). Estudos demonstraram que a hemorragia intraventricular reduz a proliferação e a sobrevivência de NSCs, contribuindo para os déficits no desenvolvimento neuropsicomotor observados em lactentes afetados (DOHARE; KIDWAI;

KAUR; SINGLA *et al.*, 2019; YAO; LI; HAN; WU *et al.*, 2023). Compreender os mecanismos celulares e moleculares por trás dessas alterações é essencial para o desenvolvimento de terapias que mitiguem os efeitos de longo prazo da hemorragia cerebral (SHARMA; AGYEMANG; BALLABH, 2022; YAO; LI; HAN; WU *et al.*, 2023).

O desenvolvimento de novos modelos para o estudo da hemorragia cerebral tem revelado que a exposição prolongada de células-tronco neurais (NSCs) à hemoglobina pode comprometer seu potencial de tronco, favorecendo a diferenciação em células gliais em detrimento da neurogênese (dados não publicados). Esse fenômeno não apenas limita a capacidade de regeneração neuronal, mas também pode contribuir para a gliose reativa e a formação de cicatrizes gliais, fatores que agravam a recuperação funcional após lesões cerebrais. Nesse contexto, a análise morfológica de NSCs surge como uma abordagem promissora, uma vez que alterações na morfologia celular frequentemente refletem mudanças em seu estado funcional e no destino celular.

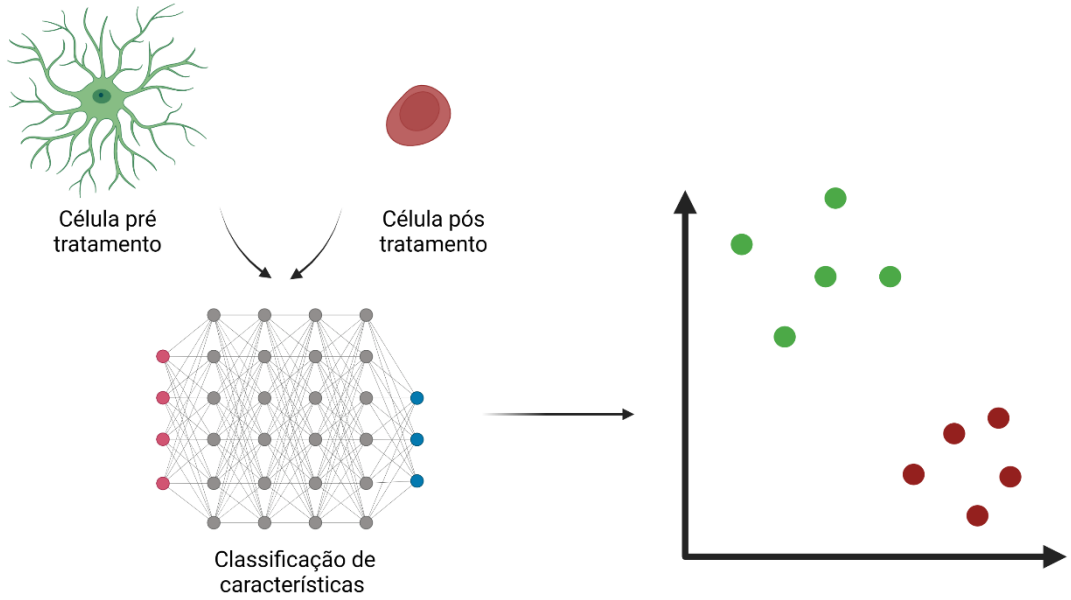


Figura 1. Os classificadores de imagens baseados em DL podem interpretar com precisão as alterações na morfologia celular na triagem de alto rendimento baseada em imagens. Esses modelos são treinados em tarefas de classificação e, em seguida, usados para extrair vetores de recursos de imagens, que podem ser agrupados para identificar novos fenótipos celulares. Criado com Biorender.com.

3 Proposta do Trabalho

A proposta deste trabalho foi avaliar como algoritmos de inteligência artificial (IA) podem ser aplicados na classificação de imagens biomédicas, com foco em identificar alterações morfológicas em células-tronco neurais (NSCs) tratadas com hemoglobina. A ideia central foi investigar a capacidade de diferentes modelos computacionais em reconhecer padrões visuais que indiquem mudanças celulares associadas ao estresse ou à diferenciação induzida, muitas vezes sutis e difíceis de serem detectadas visualmente.

Para isso, utilizamos técnicas avançadas de extração de características de imagens, como GLCM e VGG16, combinadas a algoritmos de aprendizado de máquina, incluindo SVM e MLP, para classificar as imagens de forma automatizada. O objetivo foi demonstrar como ferramentas baseadas em IA podem ser aplicadas à análise de imagens biomédicas, contribuindo para o desenvolvimento de abordagens inovadoras no estudo de alterações celulares em microambientes específicos, além de reforçar o potencial dessas técnicas na pesquisa biomédica.

4 Materiais e métodos

4.1 Aquisição de imagens

Linhagens de Células-tronco neuroepiteliais humanas (NSCs) (Control 7, indução neural dual-SMAD) foram obtidas da iPS Core Facility do Instituto Karolinska. As células foram utilizadas entre as passagens 13 e 20. Todas as linhagens foram cultivadas e passadas em DMEM/F12 Glutamax suplementado com N2 (1:100), B27 (1:1000), 10 ng/mL de bFGF e 10 ng/mL de EGF (denominado meio N2B27). As células foram cultivadas em frascos duplamente revestidos com poliornitina (PLO, 20 µg/mL) e laminina (L2020, 1:250). Os recipientes de cultura foram pré-tratados com PLO durante a noite, lavados duas vezes com DPBS (sem Ca++ e Mg++) e, em seguida, revestidos com laminina durante a noite. As NSCs foram passadas em uma proporção de 1:4 a 1:5. Resumidamente, as células foram lavadas com DPBS (sem Ca++ e Mg++) e incubadas com TrypLE™ por 3 a 4 minutos. O TrypLE™ foi neutralizado com a adição de cinco volumes de DMEM/F12 Glutamax, seguido de centrifugação a 300g e ressuspensão no meio N2B27. O meio foi completamente trocado a cada dois dias.

As células foram plaqueadas em placas de 24 poços. Parte delas foi tratada com hemoglobina humana 100 µM por 7 dias. Após esse período, as células foram fixadas com paraformaldeído (PFA) a 4% por 10 minutos à temperatura ambiente. Após duas lavagens com DPBS (com Ca++ e Mg++), as neuroesferas foram incubadas com um tampão de bloqueio contendo 10% de soro de cabra (Sigma Aldrich) e 0,1% de Triton X-100 em DPBS (com Ca++ e Mg++). A incubação com o anticorpo primário anti-nestina foi realizada em um tampão de diluição (10% do tampão de bloqueio) durante a noite a 4°C. Em seguida, as células foram incubadas com um anticorpo secundário e DAPI (4',6'-diamidino-2-fenilindol) (1:2000) em DPBS (com Ca++ e Mg++) à temperatura ambiente por uma hora. Após três lavagens em DPBS (com Ca++ e Mg++), as neuroesferas foram visualizadas usando um microscópio fluorescente Cell Observer (Zeiss, Oberkochen, Alemanha). As imagens foram processadas utilizando o software ImageJ. As imagens do canal 488 (nestin) foram salvas no formato TIFF. No total adquirimos 452 imagens do grupo “Controle” e 503 imagens do grupo “Tratado”.

4. 2 Classificação das imagens

As imagens foram classificadas utilizando diferentes abordagens implementadas no código anexado, que inclui técnicas de extração de características e algoritmos de classificação. Inicialmente, as características texturais foram extraídas das imagens utilizando a Matriz de

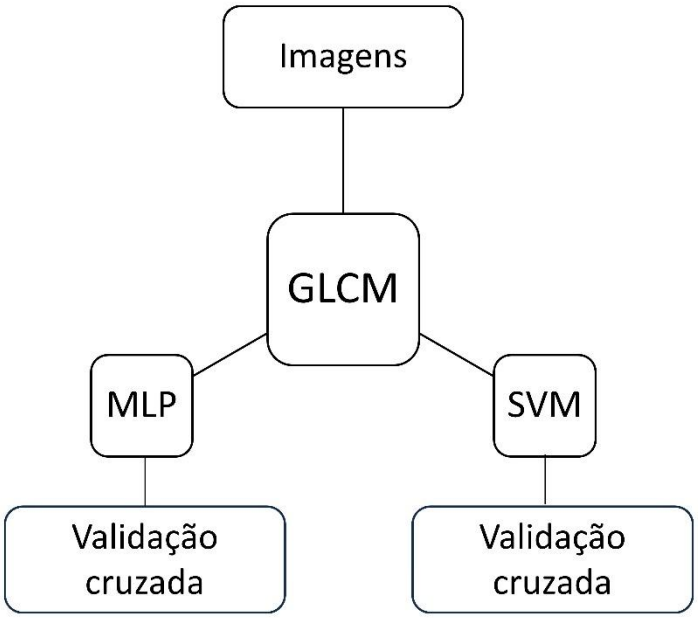
Coocorrência em Níveis de Cinza (GLCM). A análise das imagens dos grupos Controle (n = 452) e Tratado (n = 503) forneceu uma base robusta para a investigação de diferenças texturais entre as condições. Para cada imagem, foram geradas 72 características, totalizando matrizes de dimensões 452x72 para o grupo Controle e 503x72 para o grupo Tratado. Essas características representam informações sobre contraste, homogeneidade, correlação e outras propriedades relevantes das imagens.

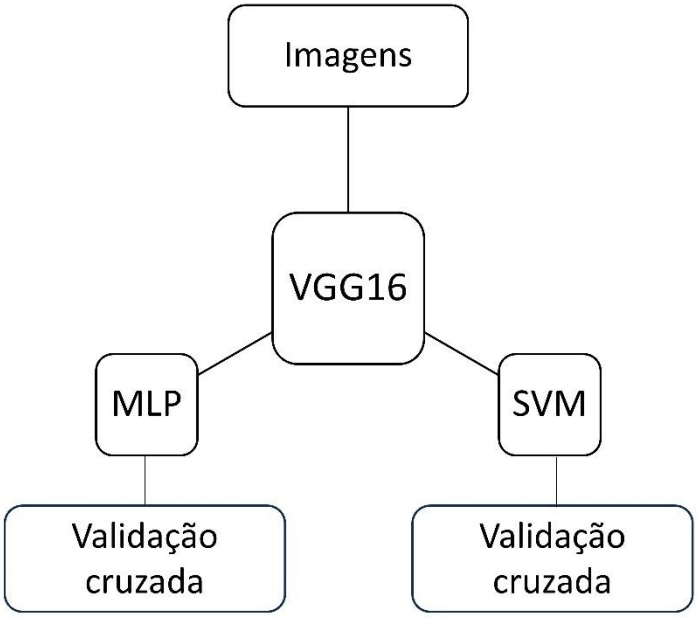
As matrizes extraídas de ambos os grupos, "Controle" e "Tratado", foram analisadas e classificadas por meio de uma Máquina de Vetores de Suporte (SVM) e de uma Rede Neural Perceptron Multicamadas (MLP).

Além disso, características das imagens também foram extraídas utilizando o modelo VGG16 pré-treinado. Para isso, a última camada do modelo foi removida, transformando a rede em um extrator de características. As informações extraídas foram então analisadas e comparadas novamente utilizando os classificadores SVM e MLP. Durante o processo, a dimensionalidade das características foi reduzida por Análise de Componentes Principais (PCA) para facilitar a visualização da separação entre os grupos e otimizar os modelos.

O código implementado foi dividido em várias etapas principais: preparação do ambiente e carregamento de dados no Google Colab, extração de características com a GLCM, análise de componentes principais, classificação com SVM e MLP, validação cruzada para verificar a robustez dos modelos, e classificação combinada utilizando o VGG16 com PCA. As etapas foram detalhadamente descritas, incluindo o uso de bibliotecas específicas como OpenCV, scikit-learn e PyTorch, além da visualização dos resultados por gráficos gerados com Matplotlib e Seaborn. Esses métodos forneceram uma abordagem integrada para a análise e classificação das imagens, permitindo avaliar a separação entre os grupos e a eficácia dos classificadores utilizados.

4.3 Pipelines





5 Resultados

5.1 Imagens

Inicialmente, ao serem adquiridas, as imagens não apresentaram visualmente nenhuma característica marcante que diferenciasse os grupos. A análise visual evidenciou a dificuldade em distinguir os dois grupos apenas a olho nu, ressaltando a necessidade de abordagens quantitativas para a identificação de diferenças potenciais. A Figura 2 ilustra exemplos representativos das imagens obtidas dos grupos Controle e Tratado, destacando a notável similaridade entre eles.

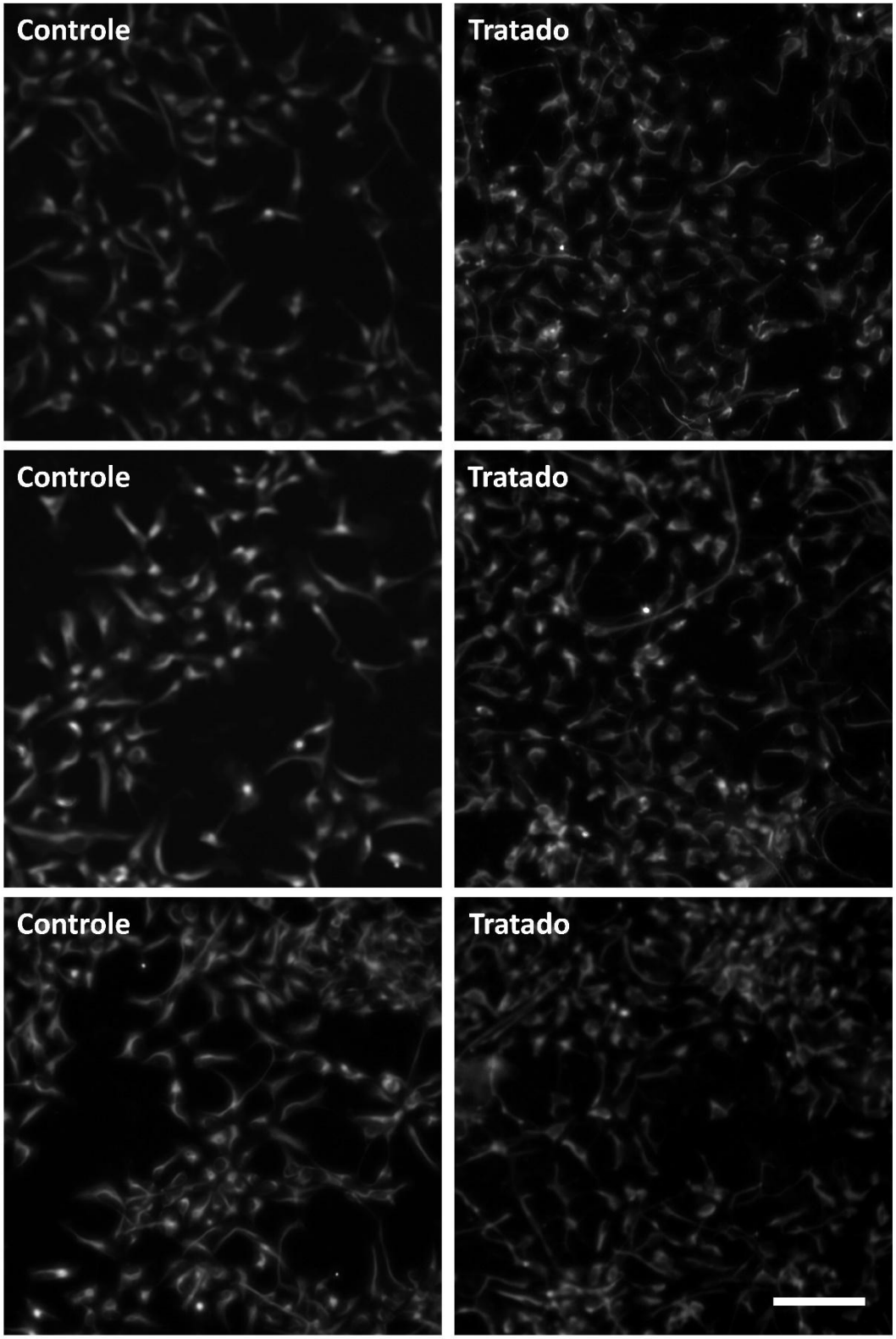


Figura 2. Exemplos de imagens dos grupos Controle e Tratado. Observa-se uma similaridade entre as imagens e dificuldade de separá-las a olho nu. Escala = 100 µm.

5.2 Características Extraídas e Análise de Padrões Texturais - GLCM

Supreendentemente, a aplicação da Análise de Componentes Principais (PCA) sobre as características extraídas pela GLCM revelou uma separação clara entre os dois grupos. A projeção nas duas primeiras componentes principais resultou em um gráfico de dispersão onde os pontos azuis (Controle) e laranja (Tratado) formaram agrupamentos distintos. Este resultado demonstra que as texturas das imagens mudaram de maneira significativa após o tratamento, refletindo alterações induzidas que afetam as propriedades morfológicas ou estruturais das amostras.

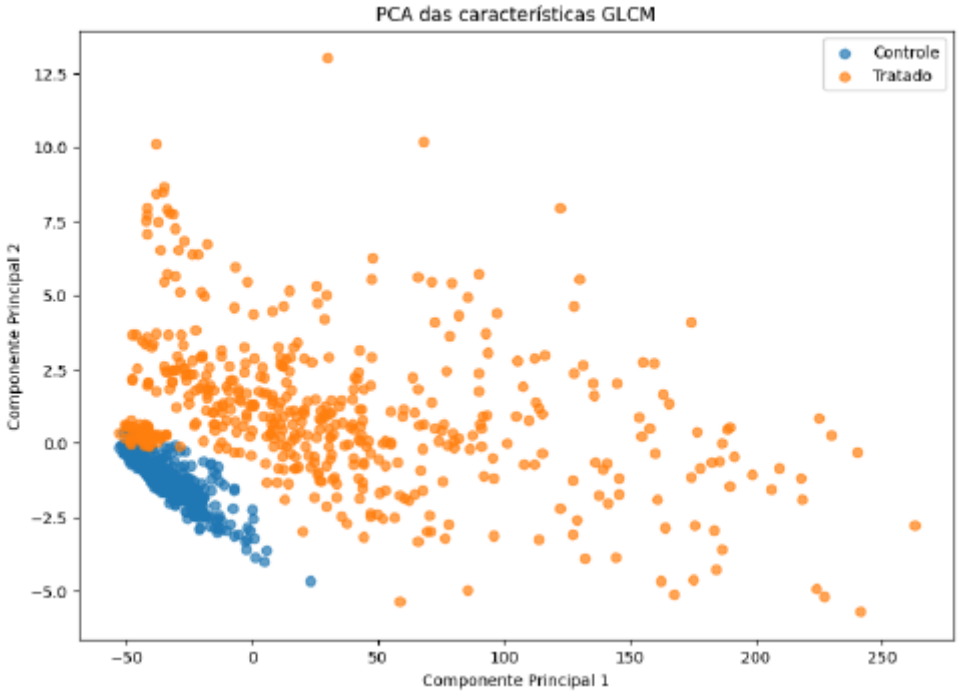
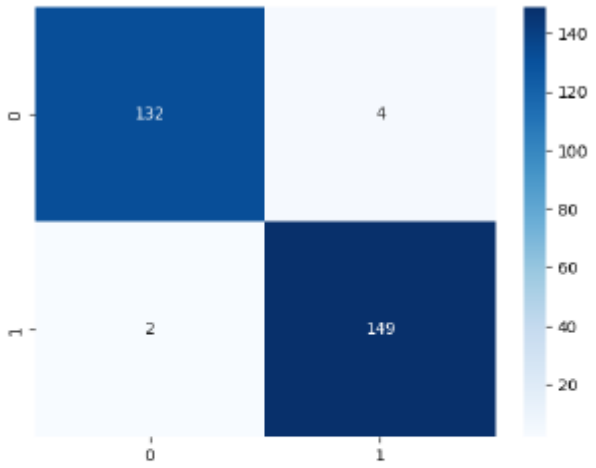


Figura 3. PCA das características extraídas da GLCM. O grupo Controle (azul) forma um único cluster enquanto o grupo Tratado (laranja) apresenta-se disperso.

5.3 Classificadores aplicados a GLCM

Os resultados do modelo SVM indicam um desempenho excelente na classificação, com métricas muito altas para ambas as classes analisadas. Para a classe Controle (classe 0), o modelo apresentou uma precisão de 99%, recall de 97% e f1-score de 98%, com 136 amostras avaliadas. Para a classe Tratado (classe 1), os resultados foram igualmente robustos, com precisão de 97%, recall de 99% e f1-score de 98%, considerando 151 amostras. A acurácia global alcançou 98%, mostrando que o modelo classificou corretamente a grande maioria das amostras. As médias macro e ponderada também refletem esse excelente desempenho, com precisão, recall e f1-score de 98%. A análise conjunta dessas métricas demonstra que o modelo

não só é eficaz em identificar corretamente as classes, como mantém um equilíbrio sólido entre precisão e sensibilidade (Figura 4). Para avaliar a robustez e a generalização do modelo, aplicamos uma validação cruzada. Dividimos os dados em 5 subconjuntos, garantindo que todas as amostras fossem utilizadas tanto para treinamento quanto para teste. Os resultados mostram que o modelo apresentou um desempenho consistente, com uma acurácia de 98% e F1-scores de 98% para ambas as classes. A precisão e o recall também foram balanceados, evidenciando a capacidade do modelo em evitar tanto falsos positivos quanto falsos negativos. Esses resultados validam a confiabilidade do modelo para tarefas de classificação binária.



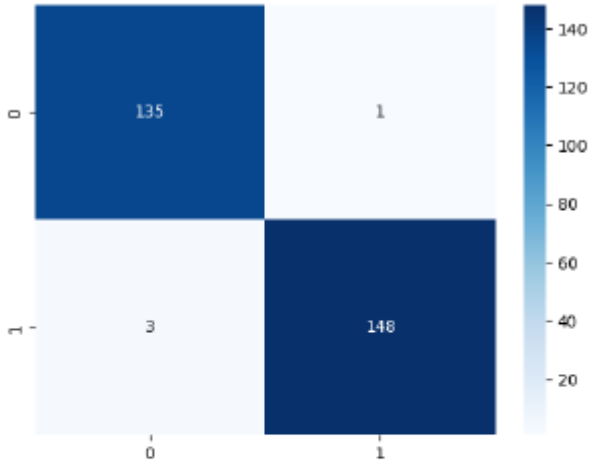
Relatório de Classificação:

	precision	recall	f1-score	support
0.0	0.99	0.97	0.98	136
1.0	0.97	0.99	0.98	151
accuracy			0.98	287
macro avg	0.98	0.98	0.98	287
weighted avg	0.98	0.98	0.98	287

Figura 4.Matriz de Confusão do classificador SVM das características extraídas pela GCLM. A matriz de confusão apresenta os valores reais (eixo vertical) contra as predições do modelo (eixo horizontal).

Já o modelo MLP superou ligeiramente o desempenho do SVM, com uma acurácia global de 99% e métricas ainda mais equilibradas entre as classes. Tanto para a classe Controle quanto para a classe Tratado, o f1-score foi de 99%, refletindo uma classificação extremamente precisa e sensível (Figura 5). A validação cruzada com 5 folds reforçou a confiabilidade do modelo, apresentando uma média de acurácia de 97,59% e um desvio padrão de 2,14%, indicando variações mínimas entre os folds. Como a acurácia dos folds individuais variou de

94,76% a 100%, investigar as pequenas discrepâncias nos folds com desempenho inferior pode ser útil para aprimorar ainda mais a estabilidade e a robustez do modelo.



Relatório de Classificação:

	precision	recall	f1-score	support
0.0	0.98	0.99	0.99	136
1.0	0.99	0.98	0.99	151
accuracy			0.99	287
macro avg	0.99	0.99	0.99	287
weighted avg	0.99	0.99	0.99	287

Figura 5.Matriz de Confusão do classificador MLP das características extraídas pela GCLM. A matriz de confusão apresenta os valores reais (eixo vertical) contra as predições do modelo (eixo horizontal).

5.4 Características Extraídas – VGG16

A rede VGG16, sem a camada final de classificação, foi utilizada como uma abordagem alternativa para a extração de características. A Análise de Componentes Principais (PCA) foi utilizada para explorar a separação das amostras entre os grupos "Controle" e "Tratado", com base nas características extraídas pela rede VGG16. O gráfico de dispersão das duas primeiras componentes principais evidenciou uma separação clara entre os dois grupos. As amostras do grupo "Controle" (em azul) concentraram-se em uma região distinta das amostras do grupo "Tratado" (em laranja) (Figura 6).

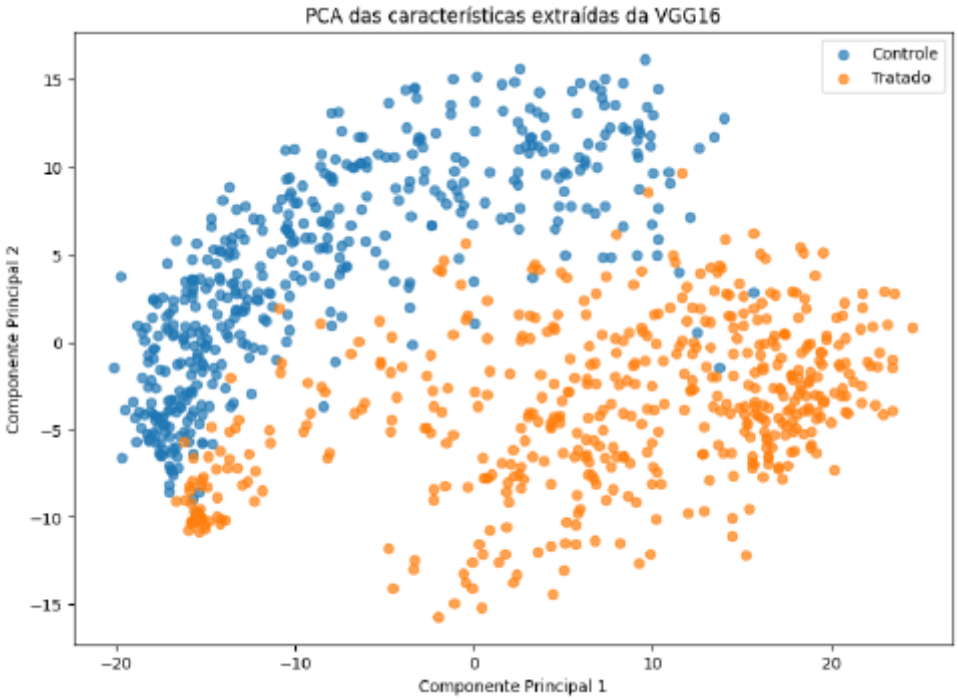
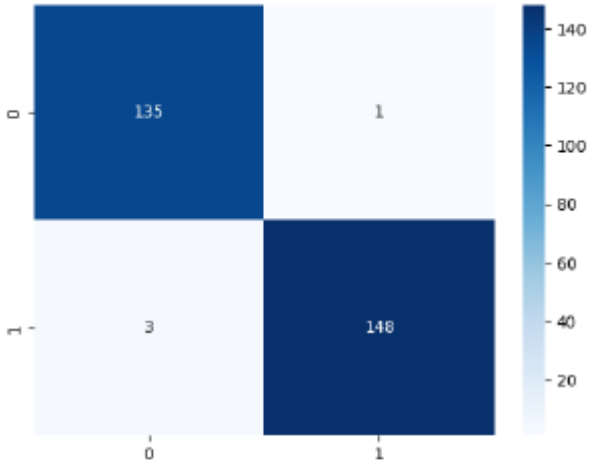


Figura 6.PCA das características extraídas da VGG16. O grupo Controle (azul) e o grupo Tratado (laranja) apresentam-se separados em direções opostas.

5.5 Classificadores aplicados à VGG16

Os resultados da classificação das características extraídas pela rede VGG16 utilizando o classificador SVM demonstraram um desempenho excepcional. O modelo atingiu uma precisão de 0,98, recall de 0,99 e F1-score de 0,99 para a classe 0 (Controle), enquanto para a classe 1 (Tratado) os valores foram 0,99, 0,98 e 0,99, respectivamente. A acurácia total foi de 99%, com médias macro e ponderada também alcançando 0,99, evidenciando um desempenho equilibrado entre as duas classes. A validação cruzada revelou acurácias de 1,0 no fold 1, 0,9948 nos folds 2 e 3, 0,9686 no fold 4 e 0,9895 no fold 5, resultando em uma média geral de 98,95% e um desvio padrão de 0,0110. Apesar de uma leve queda no desempenho no fold 4, o modelo demonstrou alta estabilidade e robustez, com uma variação mínima nos resultados ao longo dos folds. Esses resultados refletem a capacidade do SVM de classificar com precisão as amostras dos dois grupos utilizando as características extraídas pela VGG16.

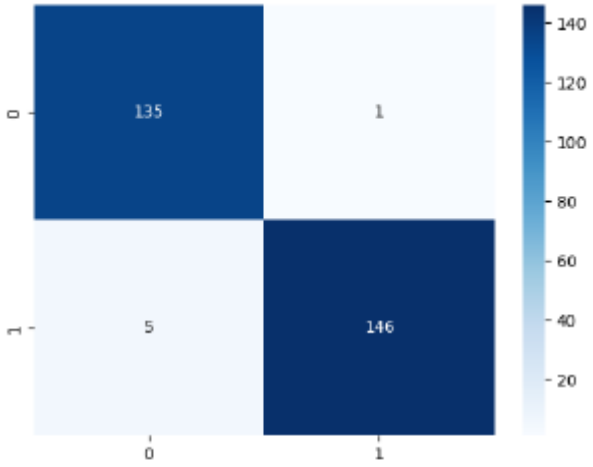


Relatório da Classificação:

	precision	recall	f1-score	support
0.0	0.98	0.99	0.99	136
1.0	0.99	0.98	0.99	151
accuracy			0.99	287
macro avg	0.99	0.99	0.99	287
weighted avg	0.99	0.99	0.99	287

Figura 7. Matriz de Confusão do classificador SVM das características extraídas pela VGG16. A matriz de confusão apresenta os valores reais (eixo vertical) contra as predições do modelo (eixo horizontal).

Já a classificação utilizando uma rede MLP treinada com as mesmas características também apresentou resultados excelentes. Para a classe 0, o modelo atingiu precisão de 0,96, recall de 0,99 e F1-score de 0,98. Para a classe 1, os valores foram de 0,99, 0,97 e 0,98, respectivamente. A acurácia total foi de 98%, com médias macro e ponderada de 0,98, indicando um desempenho igualmente equilibrado. Durante a validação cruzada, as acurácias obtidas foram de 0,9948 nos folds 1, 2, 3 e 5, e de 0,9686 no fold 4. A média geral de acurácia foi de 98,95%, com um desvio padrão de 0,0105, demonstrando uma leve variação entre os folds, mas mantendo uma consistência considerável nos resultados. A rede MLP mostrou-se altamente eficaz na tarefa de classificação, confirmando sua robustez e generalização.



Relatório da Classificação:

	precision	recall	f1-score	support
0.0	0.96	0.99	0.98	136
1.0	0.99	0.97	0.98	151
accuracy			0.98	287
macro avg	0.98	0.98	0.98	287
weighted avg	0.98	0.98	0.98	287

Figura 8. Matriz de Confusão do classificador MLP das características extraídas pela VGG16. A matriz de confusão apresenta os valores reais (eixo vertical) contra as predições do modelo (eixo horizontal).

6. Discussão

Neste trabalho, desenvolvemos pipelines para separar imagens de células-tronco neurais (NSCs) tratadas ou não com hemoglobina. Para isso empregamos dois métodos distintos para extração de características de imagens, os quais foram posteriormente utilizados na classificação por modelos de SVM ou MLP. A análise de componentes principais (PCA) desempenhou um papel central na avaliação da eficiência desses métodos, permitindo uma redução significativa da dimensionalidade dos dados e destacando as variações principais entre os grupos analisados. O gráfico de PCA para ambos os extratores revelou uma separação clara entre os grupos "Controle" e "Tratado". Isto foi surpreendente pois as análises convencionais deixavam muitas dúvidas se havia realmente uma modificação de morfologia celular. Como pode ser observado na Figura 2, a olho nu é bastante difícil reconhecer padrões entre os diferentes grupos.

Nesse sentido, a máquina se demonstrou superior ao olho humano. Os resultados demonstram que as características extraídas, tanto pela matriz de coocorrência de níveis de cinza (GLCM) quanto pela rede neural convolucional VGG16, refletem alterações texturais significativas associadas ao tratamento. Essa separação visual no PCA reforça a eficácia das abordagens empregadas, evidenciando que as características extraídas são sensíveis a mudanças texturais e podem ser utilizadas para discriminar os grupos analisados.

A comparação entre as características extraídas pelo GLCM e pela VGG16 destaca as diferenças fundamentais entre essas metodologias. O GLCM (Gray Level Co-occurrence Matrix) é uma técnica baseada na análise estatística das relações espaciais entre os níveis de cinza de pixels em uma imagem, proporcionando uma descrição intuitiva e interpretável de propriedades locais de textura, como contraste e homogeneidade (HARALICK; SHANMUGAM; DINSTEIN, 1973). Essa abordagem tem sido amplamente empregada em áreas como visão computacional, reconhecimento de padrões e análise de imagens médicas.

Para construir a GLCM, a imagem de entrada é dividida em uma grade de pixels. Em seguida, calcula-se a relação de co-ocorrência entre os níveis de cinza de cada par de pixels em uma determinada direção e distância. Essa relação é representada na matriz GLCM, na qual cada elemento reflete a frequência com que um par específico de níveis de cinza ocorre. Por permitir a captura de detalhes sutis que escapam a métodos tradicionais de processamento de imagem, a GLCM é uma ferramenta poderosa para a extração de informações texturais. Além disso, sua robustez a variações de iluminação e contraste torna-a adequada para uma ampla gama de aplicações (SO ESCOLA, 2024).

Apesar de suas vantagens, o GLCM possui algumas limitações. É particularmente sensível a ruídos na imagem, o que pode comprometer a precisão das medidas de textura. Além disso, a escolha dos parâmetros para construção da matriz, como distância e direção de co-ocorrência, influencia significativamente os resultados obtidos. Embora o GLCM seja eficaz para identificar padrões visuais específicos em imagens com baixa variabilidade estrutural, sua aplicação enfrenta desafios em imagens mais complexas, onde relações contextuais globais desempenham um papel crucial (SO ESCOLA, 2024).

O VGG16 é uma arquitetura de rede neural convolucional (CNN) desenvolvida por Karen Simonyan e Andrew Zisserman, do Visual Geometry Group da Universidade de Oxford, e apresentada em 2014 no artigo "Very Deep Convolutional Networks for Large-Scale Image Recognition". Amplamente utilizada em classificações de imagens no aprendizado profundo, a VGG16 possui 16 camadas (convolucionais e totalmente conectadas) e foi projetada para o banco de dados ImageNet (ICHI.PRO, 2024).

A entrada da rede é uma imagem RGB de 224x224 pixels, com pré-processamento simples (subtração do valor médio RGB). Suas camadas convolucionais utilizam filtros de tamanho 3x3, garantindo um maior nível de profundidade, mais não linearidades e menos parâmetros. Além disso, filtros de 1x1 são usados como transformações lineares seguidas de não linearidade. A arquitetura inclui cinco camadas de pooling máximo para redução espacial, com janelas de 2x2 pixels e passo 2. Após as camadas convolucionais, há três camadas totalmente conectadas: as duas primeiras têm 4096 canais, enquanto a última classifica 1000 categorias com uma camada softmax(ICHI.PRO, 2024).

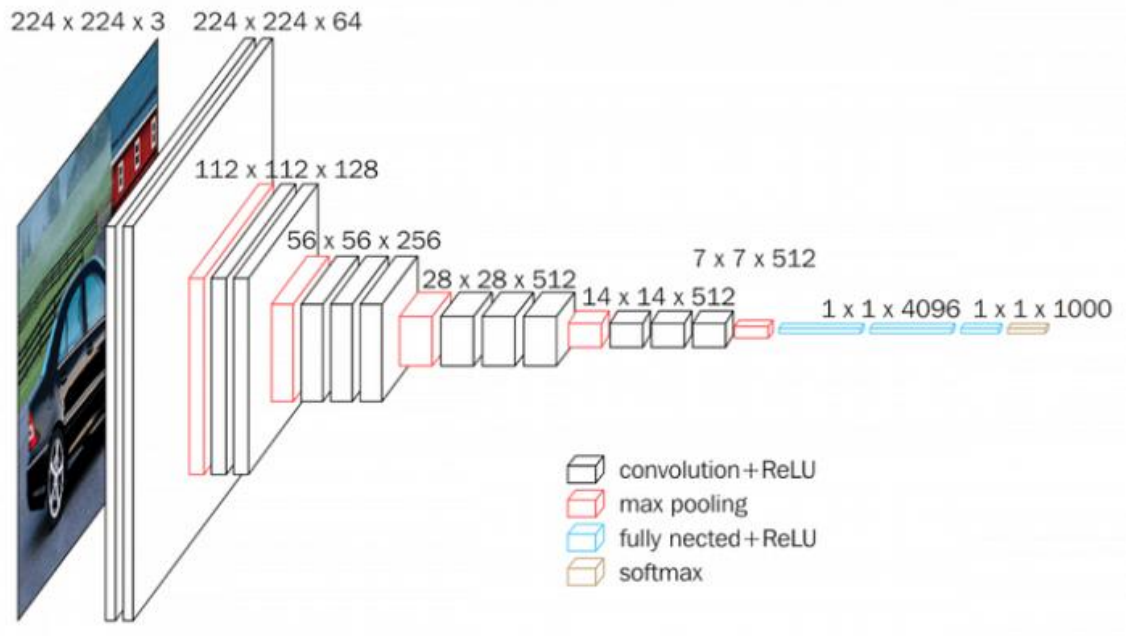


Figura 9. Arquitetura da VGG16. Extraído de https://ichi.pro/pt/o-que-e-vgg16-introducao-ao-vgg16-267001881294357. Acesso em 03 de dezembro de 2024.

Sua popularidade decorre da simplicidade de implementação e eficiência em preservar resolução espacial durante as convoluções, além de sua robustez em diversas aplicações de aprendizado profundo. A VGG16 extrai características mais abstratas e é projetada para capturar padrões globais e contextuais complexos, como formas, bordas e variações estruturais, indo além das características locais capturadas pelo GLCM. Apesar de sua menor interpretabilidade, a VGG16 se mostra extremamente eficiente em cenários que demandam análise de imagens com alta variabilidade e complexidade estrutural (JANAPA, 2020).

Quanto aos classificadores usados: as Máquinas de Vetores de Suporte (SVM) são métodos de aprendizado supervisionado usados para tarefas como classificação, regressão e detecção de outliers. Elas são amplamente aplicáveis, podendo, por exemplo, identificar células cancerígenas em imagens ou prever trajetórias com modelos de regressão. SVMs destacam-se por escolher a fronteira de decisão que maximiza a distância dos pontos mais próximos de todas as classes. Essa fronteira é chamada de hiperplano de margem máxima e garante uma separação otimizada entre as categorias. No caso de classificadores lineares, o algoritmo cria uma linha que divide duas classes, buscando a maior distância possível entre essa linha e os pontos mais próximos, conhecidos como vetores de suporte. Além disso, existem variações específicas, como o vetor de suporte de regressão (SVR), uma extensão do vetor de suporte de classificação (SVC), adaptadas para diferentes tipos de problemas. Em resumo, SVMs ajustam equações

matemáticas para fornecer respostas precisas e eficientes, sendo uma escolha robusta em diversas aplicações de aprendizado de máquina (FREECODECAMP, 2022).

O segundo classificador, o perceptron multicamadas (MLP), é uma estrutura fundamental que organiza neurônios em camadas para aprender padrões complexos. Cada neurônio recebe entradas multiplicadas por pesos, aplica uma função de ativação não linear (como ReLU ou sigmoide) e gera uma saída. A rede é treinada em um processo supervisionado, ajustando os pesos para minimizar o erro entre a saída esperada e a obtida. O backpropagation é o algoritmo que possibilita esse aprendizado. Ele calcula o erro na camada de saída e o propaga para trás, ajustando os pesos camada por camada, utilizando gradientes para orientar o treinamento na direção que reduz o erro. Isso é feito iterativamente até que o erro atinja um limite aceitável ou o número de iterações seja alcançado. Esse processo permite que redes profundas (deep learning) resolvam problemas complexos (LEITE, 2018).

Em nossa série de dados, as características extraídas por ambos os extratores foram capazes de separar os dois grupos de forma eficiente. Esse resultado foi inesperado, já que os dois grupos eram visualmente muito semelhantes. Contudo, isso indica que as células-tronco neurais, quando expostas à hemoglobina, de fato passam por um processo de estresse oxidativo e inflamatório. Dados de expressão gênica fortalecem essa conclusão, evidenciando que as células tratadas com hemoglobina perdem suas características de pluripotência e iniciam o processo de diferenciação em células da glia. As alterações morfológicas detectadas neste estudo podem representar uma manifestação direta dessa mudança no destino celular. Sob o ponto de vista biológico, essas alterações morfológicas estão possivelmente associadas a mecanismos de resposta ao estresse celular, como a remodelação do citoesqueleto e a reorganização da dinâmica da matriz extracelular. Tais processos não apenas influenciam o destino celular, mas também podem refletir uma tentativa de adaptação ao ambiente hostil provocado pelo tratamento com hemoglobina, marcando um esforço celular para preservar sua funcionalidade em condições adversas.

Uma limitação desta análise de imagens biomédicas é a falta de transparência quanto às características das imagens que são priorizadas pelo modelo para a classificação. Essa baixa interpretabilidade dificulta a compreensão das alterações morfológicas celulares mais relevantes para o desempenho do modelo.

Duas abordagens poderiam ajudar a mitigar essa limitação. A primeira seria analisar as imagens classificadas de forma equivocada ou aquelas cuja classificação foi incerta pelos algoritmos. Essa observação detalhada poderia oferecer insights valiosos sobre as características específicas que diferenciam os grupos. A segunda abordagem seria treinar o

modelo para diferenciar células-tronco neurais de astrócitos maduros e, em seguida, aplicar essa classificação ao grupo experimental. Caso o grupo tratado seja classificado predominantemente como astrócitos, isso fortaleceria a hipótese de que a exposição à hemoglobina direciona as células para um destino glial.

7 Conclusão

Os resultados deste estudo revelaram que o tratamento de células-tronco neurais (NSCs) com hemoglobina provoca alterações significativas em sua morfologia, refletidas nas características texturais das imagens. Essas alterações, detectadas por métodos computacionais, sugerem que o tratamento induz um processo inflamatório e de estresse oxidativo, corroborando evidências moleculares que indicam perda da pluripotência e transição para um fenótipo glial. A análise de componentes principais (PCA) destacou a eficácia dos extratores de características empregados, demonstrando que essas alterações podem ser capturadas de maneira robusta, mesmo quando imperceptíveis ao olho humano.

Do ponto de vista biológico, as mudanças morfológicas observadas podem estar relacionadas a mecanismos de resposta celular ao estresse, como remodelação do citoesqueleto e alterações na dinâmica da matriz extracelular. Esses processos são conhecidos por influenciar o destino celular e podem refletir uma adaptação ao ambiente adverso induzido pela hemoglobina. Além disso, a capacidade dos modelos de aprendizado de máquina em identificar padrões morfológicos específicos reforça a relevância do uso de técnicas computacionais na análise de imagens biomédicas, oferecendo novas perspectivas para investigar processos celulares complexos.

Por fim, os achados deste trabalho destacam a importância de integrar ferramentas avançadas de análise de imagens e biologia celular para compreender os efeitos do microambiente na plasticidade e no destino das células-tronco neurais. Essas abordagens têm o potencial de contribuir significativamente para a investigação de processos patológicos e para o desenvolvimento de estratégias terapêuticas mais eficazes.

REFERÊNCIAS

ALVAREZ-BUYLLA, A.; GARCIA-VERDUGO, J. M. Neurogenesis in adult subventricular zone. **J Neurosci**, 22, n. 3, p. 629-634, Feb 2002.

BEARER, E. L. Overview of image analysis, image importing, and image processing using freeware. **Curr Protoc Mol Biol**, Chapter 14, p. Unit 14.15, Aug 2003.

BUGGENTHIN, F.; BUETTNER, F.; HOPPE, P. S.; ENDELE, M. *et al.* Prospective identification of hematopoietic lineage choice by deep learning. **Nat Methods**, 14, n. 4, p. 403-406, Apr 2017.

CLARKE, S. L.; PARMESAR, K.; SALEEM, M. A.; RAMANAN, A. V. Future of machine learning in paediatrics. **Arch Dis Child**, 107, n. 3, p. 223-228, Mar 2022.

DOHARE, P.; KIDWAI, A.; KAUR, J.; SINGLA, P. *et al.* GSK3β Inhibition Restores Impaired Neurogenesis in Preterm Neonates With Intraventricular Hemorrhage. **Cereb Cortex**, 29, n. 8, p. 3482-3495, Jul 22 2019.

DRVENICA, I. T.; STANČIĆ, A. Z.; MASLOVARIĆ, I. S.; TRIVANOVIĆ, D. I. *et al.* Extracellular Hemoglobin: Modulation of Cellular Functions and Pathophysiological Effects. **Biomolecules**, 12, n. 11, Nov 17 2022.

DU, X.; CHEN, Z.; LI, Q.; YANG, S. *et al.* Organoids revealed: morphological analysis of the profound next generation in-vitro model with artificial intelligence. **Biodes Manuf**, p. 1-21, Jan 19 2023.

FREECODECAMP. *Tutorial de aprendizado de máquina: SVM*. 2022. Disponível em: https://www.freecodecamp.org/portuguese/news/tutorial-de-aprendizado-de-maquina-svm/. Acesso em: 30 nov. 2024.

GODINEZ, W. J.; HOSSAIN, I.; LAZIC, S. E.; DAVIES, J. W. *et al.* A multi-scale convolutional neural network for phenotyping high-content cellular images. **Bioinformatics**, 33, n. 13, p. 2010-2019, Jul 1 2017.

GOECKS, J.; JALILI, V.; HEISER, L. M.; GRAY, J. W. How Machine Learning Will Transform Biomedicine. **Cell**, 181, n. 1, p. 92-101, Apr 2 2020.

GU, J.; WANG, Z.; KUEN, J.; MA, L. *et al.* Recent advances in convolutional neural networks. **Pattern Recognition**, 77, p. 354-377, 2018/05/01/ 2018.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural Features for Image Classification. **IEEE Transactions on Systems, Man, and Cybernetics**, SMC-3, n. 6, p. 610-621, 1973.

HE, K.; ZHANG, X.; REN, S.; SUN, J., **Deep residual learning for image recognition**. 770-778.

HUANG, G.; LIU, Z.; VAN DER MAATEN, L.; WEINBERGER, K. Q. Densely connected convolutional networks In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. **Honolulu, HI, USA: IEEE**, p. 2261-2269, 2017.

ICHI.PRO. *O que é VGG16? Introdução ao VGG16*. Disponível em: https://ichi.pro/pt/o-que-e-vgg16-introducao-ao-vgg16-267001881294357. Acesso em: 03 dez. 2024.

JANAPA, Deepak. *Understanding VGG16: A Beginner Friendly Guide to Convolutional Neural Networks*. Medium, 16 nov. 2020. Disponível em: https://deepakjanapa.medium.com/understanding-vgg16-a-beginner-friendly-guide-to-convolutional-neural-networks-1dcf0c320e25. Acesso em: 30 nov. 2024.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. **Science**, 349, n. 6245, p. 255-260, Jul 17 2015.

KANDASWAMY, C.; SILVA, L. M.; ALEXANDRE, L. A.; SANTOS, J. M. High-Content Analysis of Breast Cancer Using Single-Cell Deep Transfer Learning. **J Biomol Screen**, 21, n. 3, p. 252-259, Mar 2016.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Communications of the ACM**, 60, n. 6, p. 84-90, 2017.

LEITE, Tiago M. *Redes Neurais, Perceptron Multicamadas e o Algoritmo Backpropagation.* Medium: Ensina*.AI*, 10 maio 2018. Disponível em: https://medium.com/ensina-ai/redes-neurais-perceptron-multicamadas-e-o-algoritmo-backpropagation-eaf89778f5b8. Acesso em: 30 nov. 2024.

LI, X.; ZHANG, L.; YANG, J.; TENG, F. Role of Artificial Intelligence in Medical Image Analysis: A Review of Current Trends and Future Directions. **Journal of Medical and Biological Engineering**, 44, n. 2, p. 231-243, 2024/04/01 2024.

MEZEI, T.; KOLCSÁR, M.; JOÓ, A.; GURZU, S. Image Analysis in Histopathology and Cytopathology: From Early Days to Current Perspectives. **Journal of Imaging**, v.10, n. 10, DOI: 10.3390/jimaging10100252.

MOEN, E.; BANNON, D.; KUDO, T.; GRAF, W. *et al.* Deep learning for cellular image analysis. **Nat Methods**, 16, n. 12, p. 1233-1246, Dec 2019.

PAWLOWSKI, N.; CAICEDO, J. C.; SINGH, S.; CARPENTER, A. E. *et al.* Automating Morphological Profiling with Generic Deep Convolutional Networks. **bioRxiv**, p. 085118, 2016.

PENG, H.; ZHOU, J.; ZHOU, Z.; BRIA, A. *et al.* Bioimage Informatics for Big Data. **Adv Anat Embryol Cell Biol**, 219, p. 263-272, 2016.

PUSHPANATHAN, K.; HANAFI, M.; MASHOHOR, S.; FAZLIL ILAHI, W. F. Machine learning in medicinal plants recognition: a review. **Artificial Intelligence Review**, 54, n. 1, p. 305-327, 2021/01/01 2021.

RIBEIRO, F. F.; XAPELLI, S. An Overview of Adult Neurogenesis. **Adv Exp Med Biol**, 1331, p. 77-94, 2021.

RIFKIND, J. M.; MOHANTY, J. G.; NAGABABU, E. The pathophysiology of extracellular hemoglobin associated with enhanced oxidative reactions. **Front Physiol**, 5, p. 500, 2014.

SHARMA, D. R.; AGYEMANG, A.; BALLABH, P. Cerebral gray matter injuries in infants with intraventricular hemorrhage. **Semin Perinatol**, 46, n. 5, p. 151595, Aug 2022.

SO ESCOLA. *O que é Gray Level Co-occurrence Matrix*. Disponível em: https://www.soescola.com/glossario/o-que-e-gray-level-co-occurrence-matrix#gsc.tab. Acesso em: 3 dez. 2024.

STĘPIEŃ, T.; TARKA, S.; CHUTORAŃSKI, D.; FELCZAK, P. *et al.* Neurogenesis in adult human brain after hemorrhage and ischemic stroke. **Folia Neuropathol**, 56, n. 4, p. 293-300, 2018.

WU, J.; JI, Y.; ZHAO, L.; JI, M. *et al.* A Mass Spectrometric Analysis Method Based on PPCA and SVM for Early Detection of Ovarian Cancer. **Comput Math Methods Med**, 2016, p. 6169249, 2016.

YAO, N.; LI, Y.; HAN, J.; WU, S. *et al.* Microglia-derived CCL20 deteriorates neurogenesis following intraventricular hemorrhage. **Experimental Neurology**, 370, p. 114561, 2023/12/01/ 2023.

ZHANG, W.; LI, R.; ZENG, T.; SUN, Q. *et al.*, **Deep model based transfer and multi-task learning for biological image analysis**. 1475-1484

Apêndce I Códigos

TCCparte1

August 31, 2024

[1]:
```
from google.colab import drive
import os

# Montar o Google Drive
drive.mount('/content/drive')

# Caminhos das pastas
controle_dir = '/content/drive/MyDrive/Imagens/Controle'
tratado_dir = '/content/drive/MyDrive/Imagens/Tratado'

# Verificar o conteúdo das pastas
print(f"Número de imagens no grupo Controle: {len(os.listdir(controle_dir))}")
print(f"Número de imagens no grupo Tratado: {len(os.listdir(tratado_dir))}")
```

```
Mounted at /content/drive
Número de imagens no grupo Controle: 452
Número de imagens no grupo Tratado: 503
```

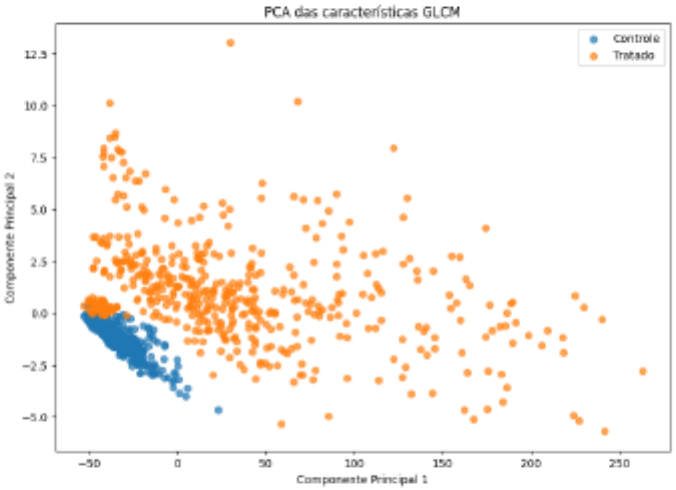
[2]:
```
import cv2
import numpy as np
from skimage.feature import graycomatrix, graycoprops
from tqdm import tqdm

def extract_glcm_features(image_path, distances, angles):
    # Carregar a imagem em escala de cinza
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    # Calcular a matriz de coocorrência em níveis de cinza (GLCM)
    glcm = graycomatrix(image, distances=distances, angles=angles,␣
 ↪symmetric=True, normed=True)
    # Extrair as propriedades da GLCM
    features = []
    for prop in ['contrast', 'dissimilarity', 'homogeneity', 'energy',␣
 ↪'correlation', 'ASM']:
        features.append(graycoprops(glcm, prop).flatten())
    return np.hstack(features)

# Definir distâncias e ângulos para GLCM
```

```
distances = [1, 2, 3]
angles = [0, np.pi/4, np.pi/2, 3*np.pi/4]

# Extrair características GLCM para todas as imagens nos grupos Controle e␣
↪Tratado
controle_features = []
tratado_features = []

for img_file in tqdm(os.listdir(controle_dir)):
    img_path = os.path.join(controle_dir, img_file)
    features = extract_glcm_features(img_path, distances, angles)
    controle_features.append(features)

for img_file in tqdm(os.listdir(tratado_dir)):
    img_path = os.path.join(tratado_dir, img_file)
    features = extract_glcm_features(img_path, distances, angles)
    tratado_features.append(features)

# Converter para numpy arrays
controle_features = np.array(controle_features)
tratado_features = np.array(tratado_features)

print(f"Características GLCM extraídas para o grupo Controle:␣
↪{controle_features.shape}")
print(f"Características GLCM extraídas para o grupo Tratado: {tratado_features.
↪shape}")
```

```
100%|      | 452/452 [01:18<00:00,  5.78it/s]
100%|      | 503/503 [01:29<00:00,  5.60it/s]
Características GLCM extraídas para o grupo Controle: (452, 72)
Características GLCM extraídas para o grupo Tratado: (503, 72)
```

[8]:
```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Concatenar as características dos dois grupos
all_features = np.vstack((controle_features, tratado_features))

# Aplicar PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(all_features)

# Separar os resultados para cada grupo
controle_pca = pca_result[:len(controle_features)]
tratado_pca = pca_result[len(controle_features):]
```

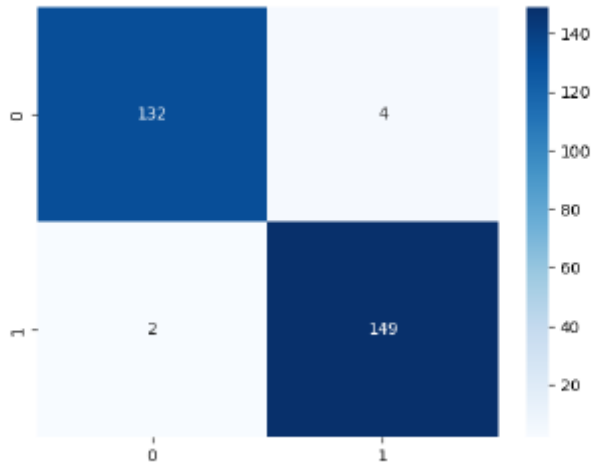
```
# Visualizar os resultados
plt.figure(figsize=(10, 7))
plt.scatter(controlo_pca[:, 0], controlo_pca[:, 1], label='Controlo', alpha=0.7)
plt.scatter(tratado_pca[:, 0], tratado_pca[:, 1], label='Tratado', alpha=0.7)
plt.title('PCA das características GLCM')
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2')
plt.legend()
plt.show()
```



[9]:
```
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# Definir os rótulos para os grupos
labels_controlo = np.zeros(len(controlo_features))  # rótulo 0 para o grupo␣
↪Controlo
labels_tratado = np.ones(len(tratado_features))    # rótulo 1 para o grupo␣
↪Tratado
```

```
# Concatenar as características e os rótulos
X = np.vstack((controle_features, tratado_features))
y = np.hstack((labels_controle, labels_tratado))

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=42, stratify=y)

# Criar e treinar a SVM
svm = SVC(kernel='linear', C=1, random_state=42)
svm.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = svm.predict(X_test)

# Avaliar o modelo
print("Matriz de Confusão:")
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.show()

print("\nRelatório de Classificação:")
print(classification_report(y_test, y_pred))
```

```
Matriz de Confusão:
```



Relatório de Classificação:

	precision	recall	f1-score	support
0.0	0.99	0.97	0.98	136
1.0	0.97	0.99	0.98	151
accuracy			0.98	287
macro avg	0.98	0.98	0.98	287
weighted avg	0.98	0.98	0.98	287

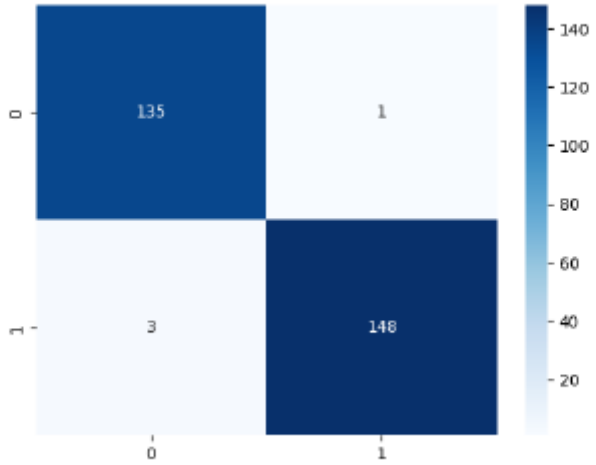
[10]:
```
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.svm import SVC

# Criar o modelo SVM
svm = SVC(kernel='linear', C=1, random_state=42)

# Configurar a validação cruzada estratificada com 5 folds
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Avaliar o modelo usando validação cruzada
```

```
cv_scores = cross_val_score(svm, X, y, cv=cv, scoring='accuracy')

# Exibir os resultados
print(f"Acurácia em cada fold: {cv_scores}")
print(f"Acurácia média: {cv_scores.mean():.4f}")
print(f"Desvio padrão da acurácia: {cv_scores.std():.4f}")
```

```
Acurácia em cada fold: [0.9895288  0.97905759 0.97382199 0.98429319 0.97382199]
Acurácia média: 0.9801
Desvio padrão da acurácia: 0.0061
```

[11]:
```
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42, stratify=y)

# Criar a MLP
mlp = MLPClassifier(hidden_layer_sizes=(100, 50), max_iter=1000,␣
 ↪random_state=42)

# Treinar a MLP
mlp.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = mlp.predict(X_test)

# Avaliar o modelo
print("Matriz de Confusão:")
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.show()

print("\nRelatório de Classificação:")
print(classification_report(y_test, y_pred))

# Validação cruzada
cv_scores = cross_val_score(mlp, X, y, cv=5, scoring='accuracy')

print(f"Acurácia em cada fold: {cv_scores}")
print(f"Acurácia média: {cv_scores.mean():.4f}")
print(f"Desvio padrão da acurácia: {cv_scores.std():.4f}")
```

```
Matriz de Confusão:
```

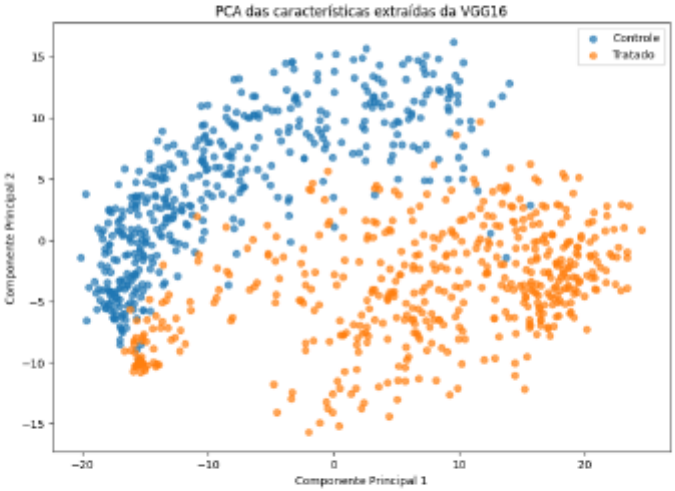



```
Relatório da Classificação:
              precision    recall  f1-score   support

         0.0       0.98      0.99      0.99       136
         1.0       0.99      0.98      0.99       151

    accuracy                           0.99       287
   macro avg       0.99      0.99      0.99       287
weighted avg       0.99      0.99      0.99       287

Acurácia em cada fold: [0.95811518 0.97382199 1.         1.         0.94764398]
Acurácia média: 0.9759
Desvio padrão da acurácia: 0.0214
```

[13]:
```
import torch
from torchvision import models, transforms
from torch import nn
from PIL import Image
import os
import numpy as np
from sklearn.decomposition import PCA
```

```
import matplotlib.pyplot as plt

# Caminhos das pastas
controle_dir = '/content/drive/MyDrive/Imagens/Controle'
tratado_dir = '/content/drive/MyDrive/Imagens/Tratado'

# Carrega o modelo VGG16 pré-treinado
model = models.vgg16(pretrained=True)

# Remove a última camada (MLP) da rede
model.classifier = nn.Sequential(*list(model.classifier.children())[:-1])

# Modo de avaliação
model.eval()

# Transforma a imagem para o formato esperado pela VGG16
preprocess = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])

def extract_features_from_directory(directory, model, preprocess):
    features_list = []
    for img_file in os.listdir(directory):
        img_path = os.path.join(directory, img_file)
        image = Image.open(img_path).convert('RGB')
        image = preprocess(image).unsqueeze(0)  # Adiciona uma dimensão de batch

        with torch.no_grad():
            features = model(image)
            features_list.append(features.numpy().flatten())

    return np.array(features_list)

# Extrair características para o grupo Controle e Tratado
controle_features = extract_features_from_directory(controle_dir, model,␣
 ↪preprocess)
tratado_features = extract_features_from_directory(tratado_dir, model,␣
 ↪preprocess)

# Concatenar as características dos dois grupos
all_features = np.vstack((controle_features, tratado_features))

# Aplicar PCA
pca = PCA(n_components=2)
```

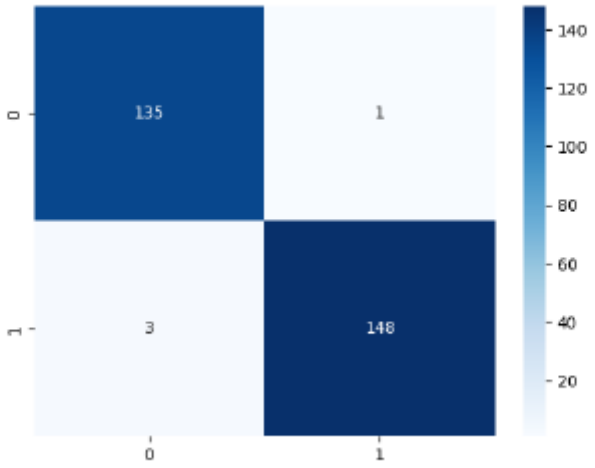
```
pca_result = pca.fit_transform(all_features)

# Separar os resultados para cada grupo
controlo_pca = pca_result[:len(controlo_features)]
tratado_pca = pca_result[len(controlo_features):]

# Visualizar os resultados
plt.figure(figsize=(10, 7))
plt.scatter(controlo_pca[:, 0], controlo_pca[:, 1], label='Controlo', alpha=0.7)
plt.scatter(tratado_pca[:, 0], tratado_pca[:, 1], label='Tratado', alpha=0.7)
plt.title('PCA das características extraídas da VGG16')
plt.xlabel('Componente Principal 1')
plt.ylabel('Componente Principal 2')
plt.legend()
plt.show()
```



[14]:
```
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
```

```
# Concatenar as características dos dois grupos (Controle e Tratado)
all_features = np.vstack((controle_features, tratado_features))

# Criar os rótulos: 0 para Controle e 1 para Tratado
labels_controle = np.zeros(len(controle_features))
labels_tratado = np.ones(len(tratado_features))
labels = np.hstack((labels_controle, labels_tratado))

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(all_features, labels,␣
↪test_size=0.3, random_state=42, stratify=labels)

# Criar e treinar a SVM
svm = SVC(kernel='linear', C=1, random_state=42)
svm.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = svm.predict(X_test)

# Avaliar o modelo
print("Matriz de Confusão:")
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.show()

print("\nRelatório de Classificação:")
print(classification_report(y_test, y_pred))

# Validação cruzada
cv_scores = cross_val_score(svm, all_features, labels, cv=5, scoring='accuracy')

print(f"Acurácia em cada fold: {cv_scores}")
print(f"Acurácia média: {cv_scores.mean():.4f}")
print(f"Desvio padrão da acurácia: {cv_scores.std():.4f}")
```

```
Matriz de Confusão:
```




```
Relatório de Classificação:
              precision    recall  f1-score   support

         0.0       0.98      0.99      0.99       136
         1.0       0.99      0.98      0.99       151

    accuracy                           0.99       287
   macro avg       0.99      0.99      0.99       287
weighted avg       0.99      0.99      0.99       287

Acurácia em cada fold: [1.         0.9947644  0.9947644  0.96858639 0.9895288 ]
Acurácia média: 0.9895
Desvio padrão da acurácia: 0.0110
```

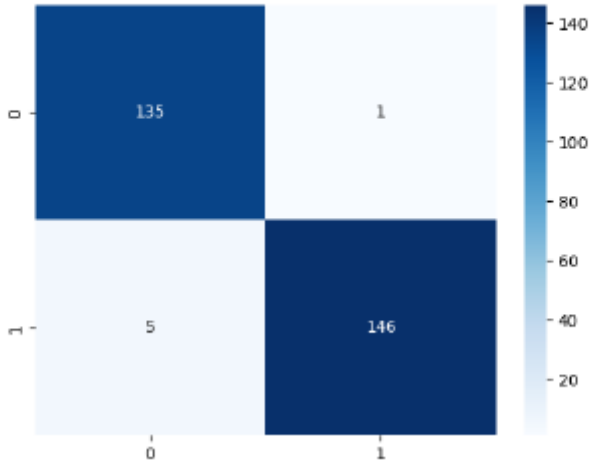
[18]:
```
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score
import numpy as np

# Criar a SVM
svm = SVC(kernel='linear', C=1, random_state=42)
```

```
# Validação cruzada com 5 folds
cv_scores = cross_val_score(svm, all_features, labels, cv=5, scoring='accuracy')

# Exibir os resultados
print(f"Acurácia em cada fold: {cv_scores}")
print(f"Acurácia média: {cv_scores.mean():.4f}")
print(f"Desvio padrão da acurácia: {cv_scores.std():.4f}")
```

```
Acurácia em cada fold: [1.         0.9947644  0.9947644  0.96858639 0.9895288 ]
Acurácia média: 0.9895
Desvio padrão da acurácia: 0.0110
```

[22]:
```
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# Dividir os dados em conjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(all_features, labels,␣
 ↪test_size=0.3, random_state=42, stratify=labels)

# Criar a MLP
mlp = MLPClassifier(hidden_layer_sizes=(100, 50), max_iter=1000,␣
 ↪random_state=42)

# Treinar a MLP
mlp.fit(X_train, y_train)

# Fazer previsões no conjunto de teste
y_pred = mlp.predict(X_test)

# Avaliar o modelo
print("Matriz de Confusão:")
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.show()

print("\nRelatório de Classificação:")
print(classification_report(y_test, y_pred))

# Validação cruzada
cv_scores = cross_val_score(mlp, all_features, labels, cv=5, scoring='accuracy')

print(f"Acurácia em cada fold: {cv_scores}")
print(f"Acurácia média: {cv_scores.mean():.4f}")
print(f"Desvio padrão da acurácia: {cv_scores.std():.4f}")
```

Matriz de Confusão:



Relatório de Classificação:

	precision	recall	f1-score	support
0.0	0.96	0.99	0.98	136
1.0	0.99	0.97	0.98	151
accuracy			0.98	287
macro avg	0.98	0.98	0.98	287
weighted avg	0.98	0.98	0.98	287

Acurácia em cada fold: [0.9947644 0.9947644 0.9947644 0.96858639 0.9947644]
Acurácia média: 0.9895
Desvio padrão da acurácia: 0.0105